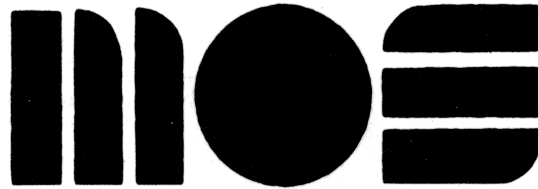
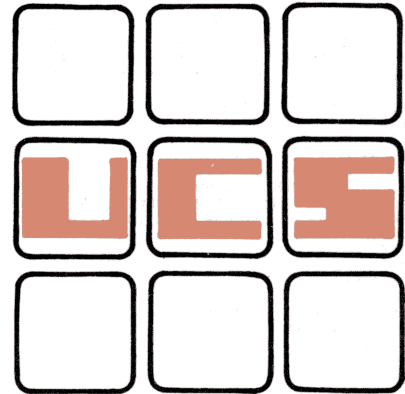


MOS TECHNOLOGY, INC.  
NORRISTOWN, PA. 19401



## MICROCOMPUTERS

### MCS6500 MICROPROCESSOR SOFTWARE SUPPORT



MOS TECHNOLOGY'S support software is now available on United Computing Systems time-sharing service. The package available provides online support to assist the microcomputer applications design engineer or programmer in program development for the MCS650X microcomputer family.

#### TO USE MOS TECHNOLOGY SUPPORT SOFTWARE:

1. Contact your local USC sales representative and request MOS TECHNOLOGY'S MCS650X Software System under user catalog number M490. Also request the UCS System Guide and the UNIEDIT manuals.
2. Order your copy of the MCS6500 Microprocessor Hardware, Programming, Simulator, And Cross Assembler manuals from: MOS Technology Inc., 950 Rittenhouse Rd., Norristown, Pa. 19401
3. Dial the appropriate telephone number supplied by your USC sales representative, sign on with your terminal, and begin entering your MCS650X microprocessor program.

#### THE SOFTWARE SUPPORT PACKAGE CONSISTS OF:

- MOS/\*\* - A text file containing the latest bulletins regarding MOS TECHNOLOGY Microprocessor Software.
- ASM/\*\* - An interactive program which builds the job control language required to submit your source code to ASM650X.
- ASM650X MCS650X Cross Assembler: the Cross Assembler is a program which translates a mnemonic or symbolic form of a computer program to machine language.
- SIM/\*\* - An interactive program which builds the job control language required to submit your simulator command file to SIM650X.

SIM650X - MCS650X Simulator. The simulator uses the command file to simulate execution of the machine language instructions created by the cross assembler in the MCS650X microprocessor.

-DMP/\*\* - ROM dump program. This program creates an output file of machine language instructions in a format suitable for MOS microcomputer loader programs.

The sample program shown in this brochure uses the UCS time-sharing system to give the user an overview of the procedure to be followed for using MOS TECHNOLOGY'S support software.

#### In brief the procedure to be followed is:

1. Create a source file using the time-sharing editor and save the file.
2. Submit the source file to the Cross Assembler by answering the questions asked by -ASM/\*\*.
3. When the Cross Assembler run is completed list the output file to obtain a listing of the assembled program.
4. Create a file of simulator commands using the time-sharing editor and save the file.
5. Submit the simulator command file and the machine language file to the simulator by answering the questions asked by -SIM/\*\*.
6. When the simulator run is completed list the output file to obtain the results of the program simulation.
7. Obtain a ROM dump object tape by answering the questions asked by -DMP/\*\*.

# 1. CREATE A SOURCE FILE.

l>pl>`plpT63

UCS 11/19/75. 09.10.41. I150  
USER NUMBER: M490010,EXAMPLE

GENERAL:  
MOS TECHNOLOGY 650X MICROPROCESSOR SOFTWARE.  
FOR THE LATEST INFORMATION TYPE -MOS/\*\*\*

MESSAGE(S) COMPLETE.

0.013 / 0.038 / 9  
READY - FOR!  
-MOS/\*\*\*

11/19/75. 09.11.22.  
PROGRAM MOS

LAST UPDATED ON 11/19/75

BULLETINS REGARDING THE MOS TECHNOLOGY MICROPROCESSOR  
SOFTWARE WILL APPEAR FROM TIME TO TIME IN THIS MANNER.

TO RUN THE 650X CROSS ASSEMBLER YOU MUST FIRST CREATE A  
SOURCE FILE. THEN ENTER -ASM/\*\*\* TO SUBMIT YOUR SOURCE FILE  
FOR BACKGROUND BATCH EXECUTION.

TO RUN THE 650X SIMULATOR YOU MUST FIRST CREATE A SIMULATOR  
COMMAND FILE AND A CROSS ASSEMBLER INTERFACE FILE. THEN TYPE -  
-SIM/\*\*\* TO SUBMIT YOUR COMMAND FILE FOR SIMULATION.

THE 650X ROM DUMP PROGRAM WILL CREATE A REFORMATED FILE  
SUITABLE FOR INPUT TO THE MOS MICROCOMPUTER LOADER PROGRAMS.  
YOU MUST HAVE CREATED AN INTERFACE FILE WITH THE CROSS  
ASSEMBLER. TO RUN THE DUMP PROGRAM ENTER -DMP650X/\*\*\*

THANK YOU.....MOS TECHNOLOGY  
RUN COMPLETE.

NEW,SAMP4  
READY - FOR!

AUT  
00100 .PAGE 'MULTIPLE BYTE ADD'  
00110 ;ADDITION OF TWO MULTIPLE PRECISION NUMBERS (BCD)  
00150 \*#0 ALLOCATE A DATA AREA IN FIRST PAGE OF MACHINE  
00170 ADDR \*\*++1  
00190 NB\*8  
00200 PP \*\*++NB  
00210 Q \*\*++NB  
00220 RES \*\*++NB  
00270 MAIN LDX #58F BEGIN MAIN ROUTINE TO TEST SUB. BCD.  
00280 TXS INITIALIZE STACK POINTER  
00290 LDX #PP  
00300 STX ADDR  
00310 JSR BCD  
00320 NOP  
00330 JMP \*-1 END OF MAIN PGM  
00360 \*=100 BEGIN SUBROUTINE  
00370 BCD LDY #NB  
00380 LDX ADDR LOADS DATA ADDRESS  
00390 CLC  
00400 SED  
00410 NEXT LDA NB-1,X  
00420 ADC 2\*NB-1,X  
00430 STA 3\*NB-1,X  
00440 DEX  
00450 DEY  
00460 BNE NEXT END OF LOOP  
00470 CLD  
00480 RTS  
00490 ABCDEFGH NOP THIS IS AN INTENTIONAL ERROR.  
00500 .END  
00510 \*DEL\*  
SAVE  
READY.

Enter proper response so that computer can determine  
your terminal's speed.

For 10 CPS enter 763  
For 15 CPS enter 863  
For 30 CPS enter T63

Enter your user number and password to log on to  
UCS system.

Indicates FORTRAN system is ready. (FORTRAN is  
automatically assigned.)

Enter -MOS/\*\*\* to obtain latest bulletins.

Indicates the end of the bulletin.

Create a new file with file name "SAMP4".

Auto line number assignment.

Assembler directive to advance listing to top of page  
and title the page "MULTIPLE BYTE ADD".

Semicolon indicates the start of a comment field.

\*# = assembler directive sets the program counter.

Sets NB equal to 8.

Reserves 8 bytes of memory for the label "PP".

Start of program labeled "MAIN"

Note that there is only one space between a line  
number and a label. There are two or more spaces  
between a line number and an instruction. Com-  
ments may begin one space after the operand.

.END assembler directive defines the end of the source  
program.

Hitting the "ESC" key ends the auto line number  
assignment. The system replies "\*\*\*DEL\*\*\*".

SAVE is the command to save the new file just creat-  
ed.

# 2. SUBMIT TO CROSS ASSEMBLER.

-ASM/\*\*\*

MOS TECHNOLOGY 650X CROSS ASSEMBLER SUBMITTOR

DO YOU WANT INSTRUCTIONS (YES OR NO) -- ? NO  
ENTER USERNUM,PASSWORD, AND PID (IF NEEDED) -- ? M490010,EXAMPLE  
DO YOU WANT TO CHANGE THE PRIORITY -- ? NO

ENTER SOURCE FILE NAME -- ? SAMP4

SAVE OUTPUT FILE (YES OR NO) -- ? YES  
ENTER OUTPUT FILE NAME -- ? OUT4

SAVE INTERFACE FILE (YES OR NO) -- ? YES  
ENTER INTERFACE FILE NAME -- ? INT4

SAVE ERROR FILE (YES OR NO) -- ? YES  
ENTER ERROR FILE NAME -- ? ERR4

SAVE DAYFILE FILE (YES OR NO) -- ? YES  
ENTER DAYFILE FILE NAME -- ? DAY4

ENTER CONTROL FILE NAME -- ? CON4

TO RUN ASSEMBLER TYPE --  
OLD,CON4  
RJE (OR RBE)

STOP.  
OLD,CON4  
READY - EXE!  
RJE

11/19/75. 09.15.45.  
PROGRAM CON4

RJE COMPLETE,ID = RJEDZQM

-ASM/\*\*\* invokes the cross assembler submittor  
software.

SOURCE file is the file containing the source code to  
be assembled.

OUTPUT file will contain the assembler listing.

INTERFACE file will contain the object code, line  
number and label information required by the sim-  
ulator.

ERROR file will contain a listing of any errors that  
occur during the assembly.

DAY file is a history of steps taken by the UCS  
system in running your job.

CONTROL file is the file of JCL built by -ASM/\*\*\*  
to run your assembly.

Submits assembly job to the UCS system.

Indicates that the job has been submitted under the  
job name "RJEDZQM".

## 3. LIST OUTPUT FILE

OLD,OUT4  
READY - EXE!  
LIS

Terminal input to list the output file "OUT4".

11/19/75. 09.18.14.  
PROGRAM OUT4

Title created by ,PAGE assembler directive.

```
+      MULTIPLE BYTE ADD      PAGE      1
0LINE  LOC      CODE      SOURCE
110      ;ADDITION OF TWO MULTIPLE PRECISION NUMBERS (BCD)
150 0000      *:=0  ALLOCATE A DATA AREA IN FIRST PAGE OF MACHINE
170 0000      ADDR *:=+1
190      NB=8
200 0001      PP *:=+NB
210 0009      Q *:=+NB
220 0011      RES *:=+NB
270 0019 A2 8F      MAIN LDX #58F BEGIN MAIN ROUTINE TO TEST SUB. BCD.
280 001B 9A      TXS  INITIALIZE STACK POINTER
290 001C A2 01      LDX #PP
300 001E 86 00      STX ADDR
310 0020 20 64 00      JSR BCD
320 0023 EA      NOP
330 0024 4C 23 00      JMP *-1  END OF MAIN PGM
360 0027      *-100  BEGIN SUBROUTINE
370 0064 A0 08      BCD LDY #NB
380 0066 A6 00      LDX ADDR  LOADS DATA ADDRESS
390 0068 18      CLC
400 0069 F8      SED
410 006A B5 07      NEXT LDA NB-1,X
420 006C 75 0F      ADC 2*NB-1,X
430 006E 95 17      STA 3*NB-1,X
440 0070 CA      DEX
450 0071 88      DEY
460 0072 D0 F6      BNE NEXT  END OF LOOP
470 0074 D8      CLD
480 0075 60      RTS
490 0076 EA EA EA  ABCDEFGH NOP  THIS IS AN INTENTIONAL ERROR.
***** ERROR ** LABEL GREATER THAN SIX CHARACTERS - NEAR COLUMN 1
500      .END
```

Program counter. (Hexadecimal)

Hexadecimal instruction, data, or value.

Program counter set to hexadecimal 64 by assembler directive \*=100.

Error line will also appear in the ERROR file.

END OF MOS/TECHNOLOGY 650X ASSEMBLY VERSION 4  
NUMBER OF ERRORS = 1, NUMBER OF WARNINGS = 0  
1 SYMBOL TABLE

The version number is changed as improvements are made to the Cross Assembler.

SYMBOL	VALUE	LINE	DEFINED	CROSS-REFERENCES
ADDR	0000	170	300	380
BCD	0064	370	310	
MAIN	0019	270	****	
NB	0008	190	200	210 220 370 410 420 430
NEXT	006A	410	460	
PP	0001	200	290	
Q	0009	210	****	
RES	0011	220	****	

RUN COMPLETE.

Note: For more detailed information refer to the MCS6500 Microprocessor Programming and Cross Assembler manuals.

## 4. CREATE SIMULATOR COMMANDS

```
NEW,ECSAMP1
READY - FOR!
AUTO
00100 SM 1 1 2 3 4 5 6 7 8
00110 SM 9 8 7 6 5 4 3 2 1
00120 DUMP 1 $18
00130 TRACE 0 $FFFF
00140 DO MAIN NEXT 3 .TIMES
00150 DUMP 1 $18
00160 EXIT
00170 *DEL*
SAVE
READY.
```

Create simulator command file called "ECSAMP1".

Starting at location 1 set consecutive memory locations to the specified values.

Dump the contents of memory from decimal 1 to hexadecimal 18.

Trace every instruction executed.

Begin simulated execution at label "MAIN" and continue until instruction at label "NEXT" has been executed 3 times.

EXIT terminates simulator run.

## 5. SUBMIT TO SIMULATOR

-SIM/\*\*\*

-SIM/\*\*\* invokes the simulator submittor software.

MOS TECHNOLOGY 650X SIMULATOR SUBMITTOR

DO YOU WANT INSTRUCTIONS (YES OR NO) -- ? NO  
ENTER USERNUM,PASSWORD, AND PID (IF NEEDED) -- ? M490010,EXAMPLE  
DO YOU WANT TO CHANGE THE PRIORITY -- ? NO

ENTER COMMAND FILE NAME -- ? ECSAMP1

COMMAND file is the file containing the simulator commands.

ENTER INTERFACE FILE NAME -- ? INT4

INTERFACE file is the interface file created by the cross assembler.

SAVE OUTPUT FILE (YES OR NO) -- ? YES

ENTER OUTPUT FILE NAME -- ? EOUT4

SAVE DAYFILE FILE (YES OR NO) -- ? YES

ENTER DAYFILE FILE NAME -- ? EDAY4

ENTER CONTROL FILE NAME -- ? ECON4

TO RUN SIMULATOR TYPE --  
OLD,ECON4  
RJE (OR RBE)

STOP.  
OLD,ECON4  
READY - EXE!  
RJE

11/19/75. 09.23.50.  
PROGRAM ECON4

RJE COMPLETE,ID = RJEDZRY

## 6. LIST SIMULATOR OUTPUT

```
OLD,EOUT4
READY - FOR!
LIST
```

Terminal commands required to list the Simulator output file.

```
11/19/75. 09.26.05.
PROGRAM EOUT4
```

```
1+++++ MOS TECHNOLOGY 650X MICROPROCESSOR SIMULATOR +++++
```

```
00100 SM 1 1 2 3 4 5 6 7 8
00110 SM 9 8 7 6 5 4 3 2 1
00120 DUMP 1 $18
```

CONTENTS OF MEMORY LOCATION AT BASE ADDRESS PLUS.....

```
BASE ADDRESS +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F
DUMP ADDR=0000 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
DUMP ADDR=0010 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00130 TRACE 0 $FFFF
00140 DO MAIN NEXT 3 .TIMES
```

Output generated as a result of the DUMP command.

```
IA LABEL OPCODE A S X Y P STATUS PC EA EO ICNT TCNT 6501 TIME
T0019 MAIN LDX A2 00 00 8F 00 90 N B 001B 001A 8F 1 2 0.
T001B TXS 9A 00 00 8F 00 90 N B 001C 001B 00 2 4 0.
T001C LDX A2 00 8F 01 00 10 B 001E 001D 01 3 6 0.
T001E STX 86 00 8F 01 00 10 B 0020 0000 01 4 9 0.
T0020 JSR 20 00 8D 01 00 10 B 0064 0064 00 5 15 0.
T0064 BCD LDY A0 00 8D 01 00 10 B 0066 0065 08 6 17 0.
T0066 LDX A6 00 8D 01 00 10 B 0068 0000 01 7 20 0.
T0068 CLC 18 00 8D 01 00 10 B 0069 0068 00 8 22 0.
T0069 SED F8 00 8D 01 00 18 BD 006A 0069 00 9 24 0.
T006A NEXT LDA B5 00 8D 01 00 18 BD 006C 0008 08 10 28 0.
T006C ADC 75 00 8D 01 00 18 BD 006E 0010 01 11 32 0.
T006E STA 95 00 8D 01 00 18 BD 0070 0018 09 12 36 0.
T0070 DEX CA 00 8D 00 00 1A BD Z 0071 0070 00 13 38 0.
T0071 DEY 88 00 8D 00 07 18 BD 0072 0071 00 14 40 0.
T0072 BNE D0 00 8D 00 07 18 BD 006A 006A 00 15 43 0.
T006A NEXT LDA B5 07 8D 00 07 18 BD 006C 0007 07 16 47 0.
T006C ADC 75 00 8D 00 07 18 BD 006E 000F 02 17 51 0.
T006E STA 95 00 8D 00 07 18 BD 0070 0017 09 18 55 0.
T0070 DEX CA 00 8D 00 07 98 N BD 0071 0070 00 19 57 0.
T0071 DEY 88 00 8D 00 06 18 BD 0072 0071 00 20 59 0.
T0072 BNE D0 00 8D 00 06 18 BD 006A 006A 00 21 62 0.
EMUL MONITOR DETECTED A WARNING-PAGE ZERO WRAP
T006A NEXT LDA B5 06 8D 00 06 18 BD 006C 0006 06 22 66 0.
+HILEV+ BREAKPOINT-NORMAL DO SEQUENCE END
00150 DUMP 1 $18
```

Trace output generated during execution of the DO sequence.

A warning to the user that his program execution caused an index register to wrap around from hexadecimal FF to 00. This may not have been planned.

Indicates normal DO sequence termination.

```
CONTENTS OF MEMORY LOCATION AT BASE ADDRESS PLUS.....
BASE ADDRESS +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F
DUMP ADDR=0000 01 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
DUMP ADDR=0010 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00160 EXIT
STOP.
RUN COMPLETE.
```

Note: For more detailed information refer to the MCS6500 Simulator manual.

## 7. PUNCH OBJECT TAPE

```
-DMP/***
```

-DMP/\*\*\* invokes the ROM dump program.

```
MOS TECHNOLOGY -- ROM DUMP
```

```
ENTER INTERFACE FILENAME ? INT4
ENTER OBJECT FILE NAME FOR OUTPUT -- ? OBJ4
OBJ4 CONTAINS OBJECT OUTPUT
```

INTERFACE file is the file created by the cross assembler.

OBJECT file is the file name the object code is to be saved in.

```
STOP.
```

```
0.135 / 0.809 / 18
OLD,OBJ4
READY - EXE!
PUNCH
```

Terminal commands required to list and punch the object tape.

Note: The paper tape punch should be turned on after the carriage return is entered.

```
;0E0019A28F9AA2018600206400EA4C230004F8
;100064A0008A60018F8B507750F9517CA88D0F607D6
;050074D860EAEAE046F
;0000030003
```

```
BYE
```

Sign-off the system by entering "BYE"

```
CT=00:20
M490010 LOG OFF. 09.30.38.
```

**brought to you by**  
**andy finkel**